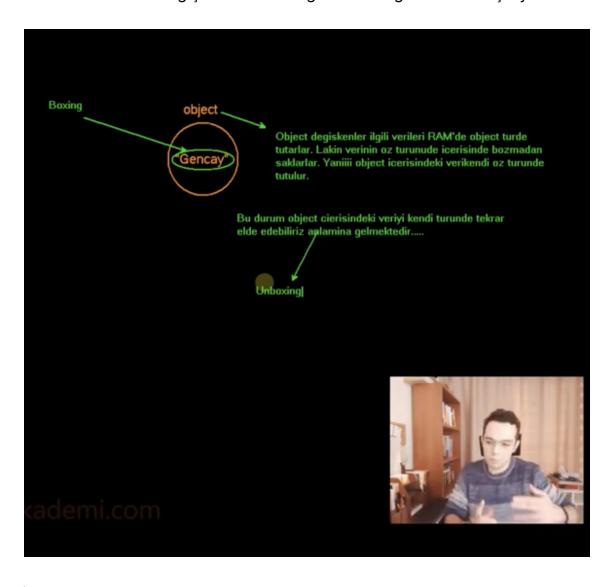
OBJECT TÜRÜ

▼ Object nedir?

- tüm türleri (değer türlü, referans türlü..)karşılayabilen bir türdür.
- Tüm türler varsayılan olarak object'ten türerler.
- Referans türlü bir değişkendir ama değer türlü değerleri de karşılayabilir.

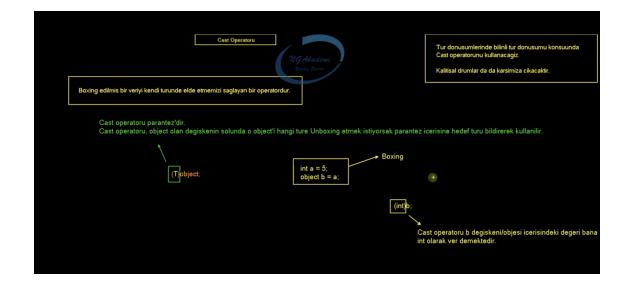


▼ Boxing

 Object türdeki bir değişkene herhangi bir türdeki değeri göndermek boxing olarak nitelendirilmektedir.

▼ Cast operatörü

- Boxing edilmiş bir veriyi (object'e atanmış veriyi) kendi türünde elde etmemizi sağlayan bir operatördür.
- Tür dönüşümlerinin bilinçli tür dönüşümü konusunda Cast operatörünü kullanacağız.
- Kalıtsal durumlarda da karşılaşcağız..
- () → cast operatörü parantezdir.
- Cast operatörü, object olan değişkenin solunda o object'in hangi türe Unboxing etmek istiyorsak parantezin içerisinde hedef turu bildirerek kullanılır.

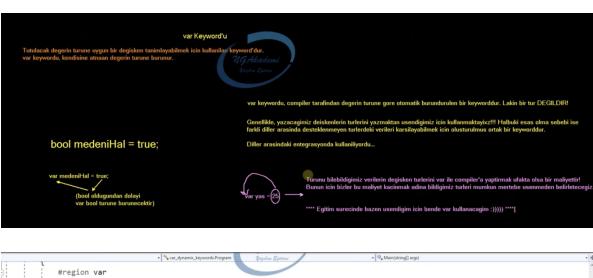


▼ Unboxing

```
    b objectType - | p | @ ; to f | ∃ % | ■ % | ■ % N N N

                           #region object
                           #region Boxing
10
11
                           //object türdeki bir değişkene herhangi bir türdeki değeri göndermek Boxing olarak nitelendirilmektedir.
12
13
14
15
16
19
20
21
22
                           object _yas = 28; //Boxing
                          Cast Operatörü
#region UnBoxing
                          #region Casting
int yas = (int) yas;//UnBoxing
Console.WriteLine(yas * 5);
23
24
25
                           #endregion
26
                           #endregion
27
                           #endregion
28
29
```

▼ var keyword'ü



```
#region var
string add = "Gencay";

var x = 3.14;

//1. var keyworduyle tanimlanan değişkenin değeri tanımlanma aşamasında verilmelidir. Verilmelidir ki türü belirleyip p
direkt ona dönüşebilsin o türde davranış sergileyebilsin...

//2. var keywordüyle tanımlanan değişkene ilk değer verildikten sonra o değerin türüne bürünceği için sonraki
p durumlarda değeri fakrlı türlerde verilemez!!! I
//3. var - object arasındaki farkı;
//var bir keyword iken object ise bir türdür. var atanan değerin türüne bürünürken, object atanan değeri Boxing
yaparak object'te dönüştürür.

#endregion

#endregion
```

▼ Dynamic keyword'ü

Var tnaımlanır tanımlanmaz, verinin türünü gösterir. Dynamikte ise bu böyle değildir, veriyi uzaktan alırız, uygulamayı build etmeden önce verininn bize hangi türde gönderildiğini anlayamayız. Runtime sırasında meydana çıkır ne olduğu.



```
0 references
static void Main(string[] args)
     #region dynamic
     //var a = 5;
     //dynamic _a = "5";
     //_a.ToStringAhmet();
     //Console.WriteLine(_a.GetType());
     //Console.WriteLine(_a * 5);
     //dynamic keywordü runtime'da türü belirleyecektir lakin kararlı davranmayacaktır.
     dynamic x = "Ahmet";
     Console.WriteLine(x.GetType());
      x = 3.14D;
                                                                Microsoft Visual Studio Debug Console
      Console.WriteLine(x.GetType());
                                                                :\Users\Gencay\Source\Repos\Application\var_dynamic_keywords\bin\Debug\netcoreapp3
2276) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debi
te when debugging stops.
Press any key to close this window . . .
      #endregion
```

